

Towards Randomized Testing of q -Monomials in Multivariate Polynomials

Shenshi Chen^{a,*}, Yaqing Chen^a, Quanhai Yang^b

^a*Department of Computer Science, University of Texas-Pan American, Edinburg, TX 78539, USA*

^b*College of Information Engineering, Northwest A&F University, Yangling, Shaanxi 712100, China*

Abstract

Given any fixed integer $q \geq 2$, a q -monomial is of the format $x_{i_1}^{s_1} x_{i_2}^{s_2} \cdots x_{i_t}^{s_t}$ such that $1 \leq s_j \leq q-1$, $1 \leq j \leq t$. q -monomials are natural generalizations of multilinear monomials. Recent research on testing multilinear monomials and q -monomials for prime q in multivariate polynomials relies on the property that Z_q is a field when $q \geq 2$ is prime. When $q > 2$ is not prime, it remains open whether the problem of testing q -monomials can be solved in some compatible complexity. In this paper, we present a randomized $O^*(7.15^k)$ algorithm for testing q -monomials of degree k that are found in a multivariate polynomial that is represented by a tree-like circuit with a polynomial size, thus giving a positive, affirming answer to the above question. Our algorithm works regardless of the primality of q and improves upon the time complexity of the previously known algorithm for testing q -monomials for prime $q > 7$.

Keywords: Algebra; complexity; multivariate polynomials; monomials; monomial testing; randomized algorithms.

1. Introduction

1.1. Background

Recently, significant efforts have been made towards studying the problem of testing monomials in multivariate polynomials [18, 22, 6, 9, 12, 10, 11, 13], with the central question consisting of whether a multivariate polynomial represented by a circuit (or even simpler structure) has a multilinear (or some specific) monomial in its sum-product expansion. This question can be answered straightforwardly when the input polynomial has been expanded into a sum-product representation, but the dilemma, though, is that obtaining such

*Corresponding author

Email addresses: `schen@broncs.utpa.edu` (Shenshi Chen), `ychen1@broncs.utpa.edu` (Yaqing Chen), `yangquanhaiyy@163.com` (Quanhai Yang)

a representation generally requires exponential time. The motivation and necessity of studying the monomial testing problem can be clearly understood from its connections to various critical problems in computational complexity as well as the possibilities of applying algebraic properties of polynomials to move forward the research on those critical problems (see, e.g., [10]).

Historically, polynomials and the studies thereof have, time and again, contributed to many advancements in theoretical computer science research. Most notably, many major breakthroughs in complexity theory would not have been possible without the invaluable roles played by low degree polynomial testing/representing and polynomial identity testing. For example, low degree polynomial testing was involved in the proof of the PCP Theorem, the cornerstone of the theory of computational hardness of approximation and the culmination of a long line of research on IP and PCP (see, Arora *et al.* [3] and Feige *et al.* [14]). Polynomial identity testing has been extensively studied due to its role in various aspects of theoretical computer science (see, for example, Kabanets and Impagliazzo [16]) and its applications in various fundamental results such as Shamir's IP=PSPACE [21] and the AKS Primality Testing [2]. Low degree polynomial representing [19] has been sought after in order to prove important results in circuit complexity, complexity class separation and subexponential time learning of Boolean functions (see, for examples, Beigel [5], Fu[15], and Klivans and Servedio [17]). Other breakthroughs in the field of algorithmic design have also been achieved by combinations of randomization and algebraization. Randomized algebraic techniques have led to the randomized algorithms of time $O^*(2^k)$ for the k -PATH problem and other problems [18, 22]. Another recent seminal example is the improved randomized $O(1.657^n)$ time algorithm for the Hamiltonian path problem by Björklund [6]. This algorithm provided a positive answer to the question of whether the Hamiltonian path problem can be solved in time $O(c^n)$ for some constant $0 < c < 2$, a challenging problem that had been open for half of a century. Björklund *et al.* further extended the above randomized algorithm to the k -path testing problem with $O^*(1.657^k)$ time complexity [7]. Very recently, those two algorithms were simplified by Abasi and Bshouty [1]. These are just a few examples and a survey of related literature is beyond the scope of this paper.

1.2. The Related Work

The problem of testing multilinear monomials in multivariate polynomials was initially exploited by Koutis [18] and then by Williams [22] to design randomized parameterized algorithms for the k -path problem. Koutis [18] initially developed an innovative group algebra approach to testing multilinear monomials with odd coefficients in the sum-product expansion of any given multivariate polynomial. Williams [22] then further connected the polynomial identity testing problem to multilinear monomial testing and devised an algorithm that can test multilinear monomials with odd or even coefficients.

The work by Chen *et al.* [9, 12, 10, 11, 13] aimed at developing a theory of testing monomials in multivariate polynomials in the context of a computational

complexity study. The goal was to investigate the various complexity aspects of the monomial testing problem and its variants.

Initially, Chen and Fu [10] proved a series of foundational results, beginning with the proof that the multilinear monomial testing problem for $\Pi\Sigma\Pi$ polynomials is NP-hard, even when each factor of the given polynomial has at most three product terms and each product term has a degree of at most 2. These results have built a base upon which further study of testing monomials can continue.

Subsequently, Chen *et al.* [13] (see, also, [12]) studied the generalized q -monomial testing problem. They proved that when $q \geq 2$ is prime, there is a randomized $O^*(q^k)$ time algorithm for testing q -monomials of degree k with coefficients $\neq 0 \pmod{q}$ in an arithmetic circuit representation of a multivariate polynomial which can then be derandomized into a deterministic $O^*((6.4p)^k)$ time algorithm when the underlying graph of the circuit is a tree.

In the third paper, Chen and Fu [9] (and [11]) turned to finding the coefficients of monomials in multivariate polynomials. Naturally, testing for the existence of any given monomial in a polynomial can be carried out by computing the coefficient of that monomial in the sum-product expansion of the polynomial. A zero coefficient means that the monomial is not present in the polynomial, whereas a nonzero coefficient implies that it is present. Moreover, they showed that coefficients of monomials in a polynomial have their own implications and are closely related to core problems in computational complexity.

1.3. Contribution and Organization

Recent research on testing multilinear monomials and q -monomials for prime q in multivariate polynomials relies on the property that Z_2 and Z_q are fields only when $q > 2$ is prime. When $q > 2$ is not prime, Z_q is no longer a field, hence the group algebra based approaches in [18, 22, 13, 12] are not applicable to cases of non-prime q . It remains open whether the problem of testing q -monomials can be solved in some compatible complexity for non-prime q . Our contribution in this paper is a randomized $O^*(7.15^k s^2(n))$ algorithm for testing q -monomials of degree k in a multivariate polynomial represented by a tree-like circuit of size $s(n)$, thus giving an affirming answer to the above question. Our algorithm works for both prime q and non-prime q as well. Additionally, for prime $q > 7$, our algorithm provides us with some substantial improvement on the time complexity of the previously known algorithm [13, 12] for testing q -monomials.

The rest of the paper is organized as follows. In Section 2, we introduce the necessary notations and definitions. In Section 3, we examine three examples to understand the difficulty to transform q -monomial testing to multilinear monomial testing. In Section 4, we propose a new method for reconstructing a given circuit and a technique to replace each occurrence of a variable with a randomized linear sum of $q - 1$ new variables. We show that, with the desired probability, the reconstruction and randomized replacements help transform the testing of q -monomials in any polynomial represented by a tree-like circuit to the

testing of multilinear monomial in a new polynomial. We design a randomized q -monomial testing algorithm in Section 5 and conclude the paper in Section 6.

2. Notations and Definitions

For $1 \leq i_1 < \dots < i_t \leq n$, $\pi = x_{i_1}^{s_1} \dots x_{i_t}^{s_t}$ is called a monomial. The degree of π , denoted by $\deg(\pi)$, is $\sum_{j=1}^t s_j$. π is multilinear, if $s_1 = \dots = s_t = 1$, i.e., π is linear in all its variables x_{i_1}, \dots, x_{i_t} . For any given integer $q \geq 2$, π is called a q -monomial if $1 \leq s_1, \dots, s_t \leq q-1$. In particular, a multilinear monomial is the same as a 2-monomial.

An arithmetic circuit, or circuit for short, is a directed acyclic graph consisting of $+$ gates with unbounded fan-ins, \times gates with two fan-ins, and terminal nodes that correspond to variables. The size, denoted by $s(n)$, of a circuit with n variables is the number of gates in that circuit. A circuit is considered a tree-like circuit if the fan-out of every gate is at most one, i.e., the underlying directed acyclic graph that excludes all the terminal nodes is a tree. In other words, in a tree-like circuit, only the terminal nodes can have more than one fan-out (or out-going edge).

Throughout this paper, the $O^*(\cdot)$ notation is used to suppress $\text{poly}(n, k)$ factors in time complexity bounds.

By definition, any polynomial $F(x_1, \dots, x_n)$ can be expressed as a sum of a list of monomials, called the sum-product expansion. The degree of the polynomial is the largest degree of its monomials in the expansion. With this expanded expression, it is trivial to see whether $F(x_1, \dots, x_n)$ has a multilinear monomial, or a monomial with any given pattern. Unfortunately, such an expanded expression is essentially problematic and infeasible due to the fact that a polynomial may often have exponentially many monomials in its sum-product expansion.

In general, a polynomial $F(x_1, \dots, x_n)$ can be represented by a circuit. This type of representation is simple and compact and may have a substantially smaller size polynomially in n , when compared to the number of all monomials in its sum-product expansion. Thus, the challenge then is to test whether $F(x_1, \dots, x_n)$ has a multilinear (or some other desired) monomial efficiently, without expanding it into its sum-product representation.

For any given $n \times n$ matrix \mathcal{A} , let $\text{perm}(\mathcal{A})$ denote the permanent of \mathcal{A} and $\det(\mathcal{A})$ the determinant of \mathcal{A} .

For any integer $k \geq 1$, we consider the group Z_2^k with the multiplication \cdot defined as follows. For k -dimensional column vectors $\vec{x}, \vec{y} \in Z_2^k$ with $\vec{x} = (x_1, \dots, x_k)^T$ and $\vec{y} = (y_1, \dots, y_k)^T$, $\vec{x} \cdot \vec{y} = (x_1 + y_1, \dots, x_k + y_k)^T$. $\vec{v}_0 = (0, \dots, 0)^T$ is the zero element in the group. For any field \mathcal{F} , the group algebra $\mathcal{F}[Z_2^k]$ is defined as follows. Every element $u \in \mathcal{F}[Z_2^k]$ is a linear addition of the form

$$u = \sum_{\vec{x}_i \in Z_2^k, a_i \in \mathcal{F}} a_i \vec{x}_i. \quad (1)$$

For any element $v = \sum_{\vec{x}_i \in Z_2^k, b_i \in \mathcal{F}} b_i \vec{x}_i$, we define

$$\begin{aligned} u + v &= \sum_{a_i, b_i \in \mathcal{F}, \vec{x}_i \in Z_2^k} (a_i + b_i) \vec{x}_i, \text{ and} \\ u \cdot v &= \sum_{a_i, b_j \in \mathcal{F}, \text{ and } \vec{x}_i, \vec{y}_j \in Z_2^k} (a_i b_j) (\vec{x}_i \cdot \vec{y}_j). \end{aligned}$$

For any scalar $c \in \mathcal{F}$,

$$cu = c \left(\sum_{\vec{x}_i \in Z_2^k, a_i \in \mathcal{F}} a_i \vec{x}_i \right) = \sum_{\vec{x}_i \in Z_2^k, a_i \in \mathcal{F}} (ca_i) \vec{x}_i.$$

The zero element in the group algebra $\mathcal{F}[Z_2^k]$ is $\mathbf{0} = \sum_{\vec{v}} 0 \vec{v}$, where 0 is the zero element in \mathcal{F} and \vec{v} is any vector in Z_2^k . For example, $\mathbf{0} = 0 \vec{v}_0 = 0 \vec{v}_1 + 0 \vec{v}_2 + 0 \vec{v}_3$, for any $\vec{v}_i \in Z_2^k$, $1 \leq i \leq 3$. The identity element in the group algebra $\mathcal{F}[Z_2^k]$ is $\mathbf{1} = 1 \vec{v}_0 = \vec{v}_0$, where 1 is the identity element in \mathcal{F} . For any vector $\vec{v} = (v_1, \dots, v_k)^T \in Z_2^k$, for $i \geq 0$, let $(\vec{v})^i = (iv_1, \dots, iv_k)^T$. When the field \mathcal{F} is Z_2 with respect to $(\text{mod } 2)$ operation, for any $x, y \in Z_2$, xy and $x + y$ stands for $xy \pmod{2}$ and $x + y \pmod{2}$, respectively. In particular, in the group algebra $Z_2[Z_2^k]$, for any $\vec{z} \in Z_2^k$ we have $(\vec{v})^0 = (\vec{v})^2 = \vec{v}_0$.

3. *q*-Monomials, Multilinear Monomials and Plus Gates

As we pointed out before, group algebra based algorithms [18, 22, 13, 12] cannot be called upon to test *q*-monomials when *q* is not prime, because Z_q is not a field. Hence, in such a case the algebraic foundation for applying those algorithms is no longer available.

It seems quite hopeful that there might be a way to transform the problem of testing *q*-monomials into the problem of testing multilinear monomials and thus utilize the existing techniques for the latter problem to solve the former problem. One plausible strategy to accomplish such a transformation is to replace each variable x in a given multivariate polynomial by a sum $y_1 + y_2 + \dots + y_{q-1}$ of $q - 1$ new variables. Ideally, such replacements should result in a multilinear monomial in the new polynomial that corresponds to the given *q*-monomial in the original polynomial and vice versa, thereby allowing the multilinear monomial testing algorithm based on some group algebra over a field of characteristic 2 [18, 22] to be adopted for the testing of multilinear monomials in the new polynomial. Unfortunately, some careful analysis will reveal that this approach has, as exhibited in Example 1, a profound technical barrier that prevents us from applying those multilinear monomial testing algorithms.

Example 1. Consider a simple 4-monomial $\pi = x^3$ of degree 3. Replacing x with $y_1 + y_2 + y_3$ in π results in

$$\begin{aligned} r(\pi) &= (y_1 + y_2 + y_3)^3 \\ &= y_1^3 + y_2^3 + y_3^3 + 3y_1^2y_2 + 3y_1y_2^2 + 3y_2^2y_3 + 3y_2y_3^2 + 3y_1^2y_3 + 3y_1y_3^2 \\ &\quad + 6y_1y_2y_3. \end{aligned}$$

$r(\pi)$ has one and only one degree 3 multilinear monomial $\pi' = y_1y_2y_3$. It is unfortunate that the coefficient $c(\pi')$ of π' is 6, an even number. When applying the group algebra based multilinear monomial testing algorithms to $r(\pi)$ over the field Z_2 with respect to (mod 2) operation, the even coefficient $c(\pi')$ will help eliminate π' from $r(\pi)$. Hence, we are unable to find the existence of any multilinear monomials in the sum-product expansion of $r(\pi)$.

Knowing that the above example can be generalized to arbitrary q -monomials for $q > 2$, we have to design an innovative replacement technique so that certain multilinear monomials in the new polynomial will survive the elimination by the (mod 2) operation over Z_2 , or by the characteristic 2 property over any field of characteristic 2. Specifically, we have to ensure, with complete or desired probabilistic certainty, that a given q -monomial π with coefficient $c(\pi)$ in the original polynomial will correspond to one or a list of "distinguishable" multilinear monomials with odd coefficients in the derived polynomial, regardless of the parity of $c(\pi)$.

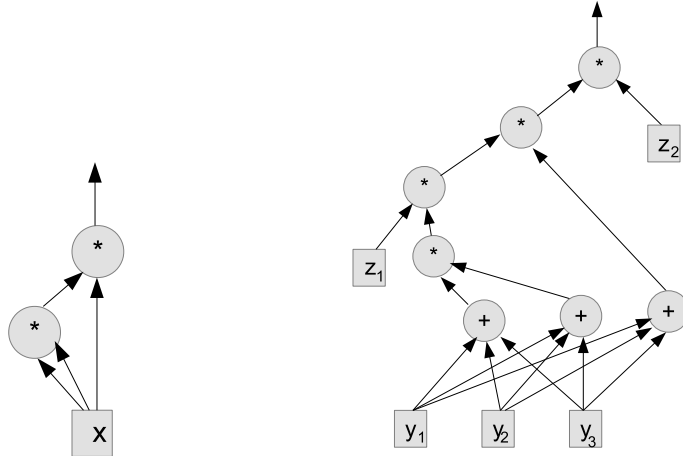


Fig. 1. A Circuit for $\pi = x^3$

Fig. 2. The Expanded Circuit for $\pi = x^3$

When group algebraic elements are selected to replace variables in the input polynomial, the polynomial might become zero due to mutual annihilation of the results from a list of multilinear monomials with odd coefficients. Koutis [18] proved that when those group algebraic elements are uniform random, with a probability at least $\frac{1}{4}$, the input polynomial that has multilinear monomials with odd coefficients will not become zero, even if mutual annihilation of the results from a list of multilinear monomials with odd coefficients may happen.

Williams [22] introduced a new variable for each \times gate in the representative circuit for the input polynomial that can help avoid the aforementioned mutual annihilation. In essence, the new variables added for the \times gates can help generate one or a list of "distinguishable" multilinear monomials with odd coefficients in the derived polynomial, no matter whether the coefficient of the original multilinear monomial is even or odd. However, this approach cannot help resolve the q -monomial testing problem, due to possible implications of $+$ gates.

In order to understand the above situation, let us examine Example 1 again. Following Williams's algorithm, we first reconstruct the circuit in Figure 1. The expanded circuit, after the replacement of x by $y_1 + y_2 + y_3$ along with the addition of new variables z_1 and z_2 for the two respective $+$ gates, is shown in Figure 2. The coefficient for the only multilinear monomial $y_1y_2y_3$ produced by the new circuit is $6z_1z_2$, which is even and thus helps annihilate $y_1y_2y_3$ with respect to $(\text{mod } 2)$ operation or in general the characteristic 2 property of the underlying field.

The following two examples provide us with more evidences that there are technical difficulties in dealing with possible implications of $+$ gates.

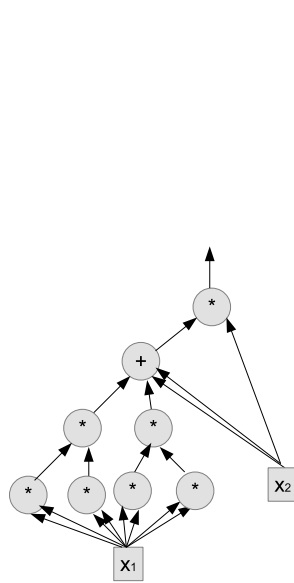


Fig. 3. A Circuit for $F(x_1, x_2)$

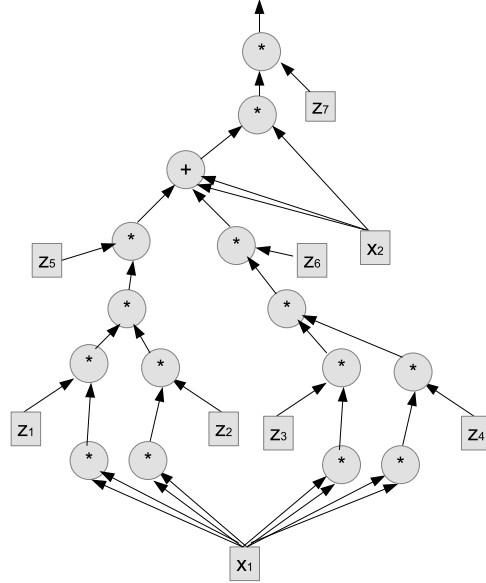


Fig. 4. The Reconstructed Circuit for $F(x_1, x_2)$

Example 2. Let $F(x_1, x_2) = 2x_1^4x_2 + 2x_2^2$ as represented by the circuit in Figure 3. F has one 5-monomial $\pi_1 = x_1^4x_2$ and one 3-monomial $\pi_2 = x_2^2$, each of which has a coefficient 2.

When one follows the approach by Williams [22] to add, for each \times gate in Figure 3, a new \times gate that multiplies the output of this gate with a new

variable, then one obtains a new circuit in Figure 4 that computes

$$F'(z_1, z_2, \dots, z_7, x_1, x_2) = z_1 z_3 z_5 z_7 x_1^4 x_2 + z_3 z_4 z_6 z_7 x_1^4 x_2 + 2z_7 x_2^2.$$

Although $2x_1^4 x_2$ in F is spilt into two distinguishable occurrences that have respective unique coefficients $z_1 z_3 z_5 z_7$ and $z_3 z_4 z_6 z_7$, yet $2x_2^2$ in F corresponds to $2z_7 x_2^2$ that has an even coefficient $2z_7$.

In particular, the implications of $+$ gates on testing multilinear monomials can be seen from the following example.

Example 3. Let $G(x_1, x_2, x_3) = 2x_1^2 x_3 + 2x_2 x_3$. Changing the terminal node x_2 to x_3 for the top \times gate in Figure 3 (respectively, for the top second \times gate in Figure 4) gives a circuit to compute G (respectively, G').

Like in Example 2, $G'(z_1, z_2, z_3, x_1, x_2, x_3) = z_1 z_3 z_5 z_7 x_1^4 x_3 + z_3 z_4 z_6 z_7 x_1^4 x_3 + 2z_7 x_2 x_3$. Here, $2x_1^4 x_3$ is spilt into two distinguishable occurrences that have unique coefficients $z_1 z_3 z_5 z_7$ and $z_3 z_4 z_6 z_7$, respectively. However, the only multilinear monomial $2x_2 x_3$ in G corresponds to $2z_7 x_2 x_3$ that has an even coefficient $2z_7$. Therefore, this multilinear monomial cannot be detected by Williams' algorithm.

Example 3 exhibits that there is a flaw in the circuit reconstruction by Williams [22]: Introducing a new variable to multiply the output of every \times gate is not sufficient to overcome the difficulty that may possibly be caused by $+$ gates.

4. Circuit Reconstruction and A Transformation

In this section, we shall design a new method to reconstruct a given circuit and a randomized variable replacement technique so that we can transform, with some desired success probability, the testing of q -monomials to the testing of multilinear monomials.

To simplify presentation, we assume from now on through the rest of the paper that if any given polynomial has q -monomials in its sum-product expansion, then the degrees of those multilinear monomials are at least k and one of them has exactly a degree of k . This assumption is feasible, because when a polynomial has q -monomials of degree $< k$, e.g., the least degree of those is ℓ with $1 \leq \ell < k$, then we can multiply the polynomial by a list of $k - \ell$ new variables so that the resulting polynomial will have q -monomials with degrees satisfying the aforementioned assumption.

4.1. Circuit Reconstruction

For any given polynomial $F(x_1, x_2, \dots, x_n)$ represented by a tree-like circuit \mathcal{C} of size $s(n)$, we first reconstruct the circuit \mathcal{C} in three steps as follows.

Eliminating redundant $+$ gates. Starting with the root gate, check to see whether a $+$ gate receives input from another $+$ gate. If a $+$ gate g receives input from a $+$ gate f , which receives inputs from gates f_1, f_2, \dots, f_s and/or

terminal nodes u_1, u_2, \dots, u_t , then delete f and let the gate g to receive inputs directly from f_1, f_2, \dots, f_s and/or u_1, u_2, \dots, u_t . Repeat this process until there are no more $+$ gates receiving input from another $+$ gate.

Note that we consider tree-like circuits only. Since each gate of such a circuit has at most one output, the above eliminating process will not increase the size of the circuit.

Duplicating terminal nodes. For each variable x_i , if x_i is the input to a list of gates g_1, g_2, \dots, g_ℓ , then create ℓ terminal nodes u_1, u_2, \dots, u_ℓ such that each of them represents a copy of the variable x_i and g_j receives input from u_j , $1 \leq j \leq \ell$.

Let \mathcal{C}^* denote the reconstructed circuit after the above two reconstruction steps. Since the original circuit \mathcal{C} is tree-like, the underlying graph of \mathcal{C}^* , including all the terminal nodes, is a tree. Such a tree structure implies the following simple facts:

- There is no duplicated occurrence of any input variable along any path from the root to a terminal node.
- Every occurrence of each variable x_i in the sum-product expansion of F is represented by a terminal node for x_i .
- The size of the new circuit is at most $ns(n)$.
- Any $+$ gate will receive input from \times gates and/or terminal nodes.

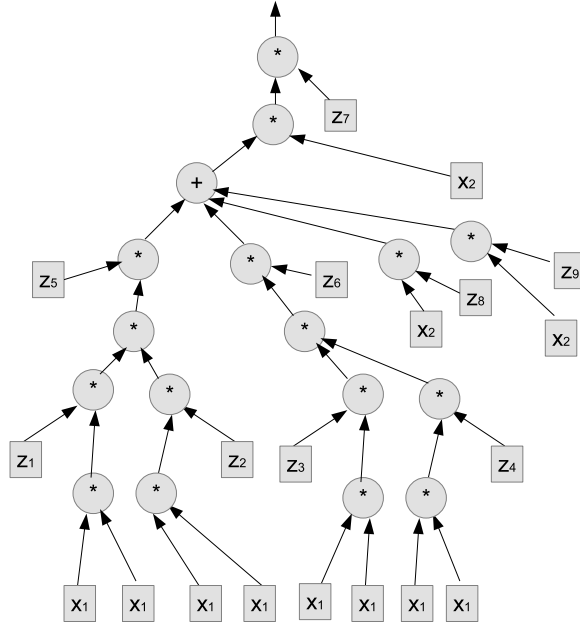


Fig. 5. The New Circuit for $F(x_1, x_2) = 2x_1^4x_2 + 2x_2^2$

Adding new variables for \times gates and for those terminal nodes that directly connect to $+$ gates. Having completed the reconstruction for

\mathcal{C}^* , we then expand it to a new circuit \mathcal{C}' as follows. For each \times gate g_i in \mathcal{C}^* , we attach a new \times gate g'_i that multiplies the output of g_i with a new variable z_i , and feed the output of g'_i to the gate that reads the output of g_i . Here, the way of introducing new variables for \times gates follows what is done by Williams in [22]. However, in addition to these new z -variables, we may need to introduce additional variables for $+$ gates. Specifically, for each $+$ gate f that receives inputs from terminal nodes u_1, u_2, \dots, u_t , we add a \times gate f_j and have it to receive inputs from u_j and a new variable z_j and then feed its output to f , $1 \leq j \leq t$. Note that f may receive input from \times gates but no new gates are needed for those gates with respect to f .

Assume that a list of h new z -variables z_1, z_2, \dots, z_h have been introduced into the circuit \mathcal{C}' . Let $F'(z_1, z_2, \dots, z_h, x_1, x_2, \dots, x_n)$ be the new polynomial represented by \mathcal{C}' .

In Figure 5, we show the reconstructed circuit for the one in Figure 3 that represents $F(x_1, x_2) = 2x_1^4x_2 + 2x_2^2$. By this new circuit,

$$F'(z_1, z_2, \dots, z_9, x_1, x_2) = z_1z_2z_5z_7x_1^4x_2 + z_3z_4z_6z_7x_1^4x_2 + z_7z_8x_2^2 + z_7z_9x_2^2.$$

As expected, not only is $2x_1^4x_2$ in F split into two distinguishable occurrences that have unique coefficients $z_1z_2z_5z_7$ and $z_3z_4z_6z_7$, but also $2x_2^2$ in F is split into two distinguishable occurrences that have unique coefficients z_7z_8 and z_7z_9 . Notably, those four coefficients are multilinear monomials of z -variables and each has an odd scalar coefficient 1.

Lemma 1. *$F(x_1, x_2, \dots, x_n)$ has a monomial π of degree k in its sum-product expansion if and only if there is a monomial $\alpha\pi$ in the sum-product expansion of $F'(z_1, z_2, \dots, z_h, x_1, x_2, \dots, x_n)$ such that α is a multilinear monomial of z -variables with degree $\leq 2k - 1$. Furthermore, if F' has two products $\alpha_1\pi$ and $\alpha_2\pi$ in its sum-product expansion, then we have $\alpha_1 \neq \alpha_2$, where α_1 and α_2 are products of z -variables; and any two different monomials of x -variables in F' will have different coefficients that are products of z -variables.*

PROOF. By the reconstruction processes, \mathcal{C}^* computes exactly the same polynomial F . If F has a monomial π of degree k , then let \mathcal{T} be the subtree of \mathcal{C}^* that generates the monomial π , and \mathcal{T}' be the corresponding subtree of \mathcal{T} in \mathcal{C}' . By the way the new z -variables are introduced, the monomial generated by \mathcal{T}' is $\alpha\pi$ with α as the product of all the z -variables added to \mathcal{T} to yield \mathcal{T}' . Since π has degree k , \mathcal{T} has $k - 1$ many \times gates. So, \mathcal{T}' has $k - 1$ new \times gates along with $k - 1$ many new z -variables that are added with respect to those \times gates in \mathcal{T} . In addition, \mathcal{T}' has k terminal nodes representing k individual copies of x -variables in π . When such a terminal node is connected to a $+$ gate, then a new \times gate is added along with a new z -variable. Thus, the terminal nodes in \mathcal{T}' can contribute at most k additional z -variables. Therefore, the degree of α is at most $2k - 1$. Since all those z -variables are distinct, α is multilinear.

If F' has a monomial $\alpha\pi$ such that α is a product of z -variables and π is a product of x -variables, then let \mathcal{M}' be the subtree of \mathcal{C}' that generates $\alpha\pi$. According to the construction of \mathcal{C}^* and \mathcal{C}' , removing all the z -variables along

with the newly added \times gates from \mathcal{M}' will result in a subtree \mathcal{M} of \mathcal{C}^* that generates π . Thereby, π is a monomial in F .

Assume that F' has $\alpha_1\pi$ and $\alpha_2\pi$ in its sum-product expansion, where α_1 and α_2 are products of z -variables. Let \mathcal{T}'_1 and \mathcal{T}'_2 be the two subtrees in \mathcal{C}' that generate $\alpha_1\pi$ and $\alpha_2\pi$, respectively. Since each of such subtrees in \mathcal{C}' can be used once to generate one product in the sum-product expansion of F' , we have $\mathcal{T}'_1 \neq \mathcal{T}'_2$. Let \mathcal{T}_1 and \mathcal{T}_2 be the two respective subtrees of \mathcal{T}'_1 and \mathcal{T}'_2 in \mathcal{C}^* . By the ways of circuit reconstruction and introduction of new z -variables, $\mathcal{T}'_1 \neq \mathcal{T}'_2$ implies $\mathcal{T}_1 \neq \mathcal{T}_2$. Note that \mathcal{T}_1 and \mathcal{T}_2 generates the same π . There are two cases for \mathcal{T}_1 and \mathcal{T}_2 to differ: either \mathcal{T}_1 and \mathcal{T}_2 differ at a \times gate g , or they have the same \times gates but differ at a terminal node u . In the former case, the z -variables added with respect to g will make α_1 and α_2 different. In the latter case, we assume without loss of generality that \mathcal{T}_1 has a terminal node u but \mathcal{T}_2 does not. In this case, the parent node u' of u has to be a $+$ gate. Hence, a new z -variable is added for the new \times gate between u' and u . Therefore, this new z -variable makes α_1 and α_2 different.

Now, consider that F' has two monomials $\alpha\pi$ and $\beta\phi$ such that, π and ϕ are products of x -variables and α and β are products of z -variables. Let \mathcal{H}'_1 and \mathcal{H}'_2 be the subtrees in \mathcal{C}' that generate $\alpha\pi$ and $\beta\phi$, respectively. Again, according to the construction of \mathcal{C}^* and \mathcal{C}' , removing all the z -variables along with the newly added \times gates from \mathcal{H}'_1 and \mathcal{H}'_2 will result in two subtrees \mathcal{H}_1 and \mathcal{H}_2 of \mathcal{C}^* that generate π and ϕ , respectively. When $\pi \neq \phi$, \mathcal{H}_1 and \mathcal{H}_2 are different subtrees. Following a similar analysis in the above paragraph for \mathcal{T}_1 and \mathcal{T}_2 to be different, we have $\alpha \neq \beta$. Also, since the z -variables in α corresponds to \times gates in \mathcal{H}'_1 that do not repeat themselves because \mathcal{H}'_1 is a tree, α is multilinear. Similarly, β is also multilinear.

Combining the above analysis completes the proof for the lemma.

4.2. A Transformation

In order to present the technique to transform the testing of q -monomials to the testing of multilinear monomials, we introduce one more definition related to variable replacements.

Definition 1. Let $q \geq 2$ be a fixed integer. Let $\pi = x^s$ for $1 \leq s \leq q-1$. Consider

$$r(\pi) = \prod_{i=1}^s (c_{i1}y_1 + c_{i2}y_2 + \cdots + c_{i(q-1)}y_{q-1}),$$

where c_{ij} are constants and y_j are new variables, $1 \leq i \leq s$ and $1 \leq j \leq q-1$. For $1 \leq s \leq q-1$, let $\pi' = y_1y_2 \cdots y_s$. Define the coefficient matrix of π' with respect to $r(\pi)$ as

$$\mathcal{C}[\pi', r(\pi)] = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1s} \\ c_{21} & c_{22} & \cdots & c_{2s} \\ & & \cdots & \\ c_{s1} & c_{s2} & \cdots & c_{ss} \end{pmatrix}.$$

Transformation: For any given n -variate polynomial $F(x_1, x_2, \dots, x_n)$ represented by a circuit \mathcal{C} , we first carry out the circuit reconstruction as addressed in Subsection 4.1 to obtain a new circuit \mathcal{C}' and let $F'(z_1, z_2, \dots, z_h, x_1, x_2, \dots, x_n)$ be the new polynomial represented by \mathcal{C}' . The transformation through replacing x -variables works as follows: For each variable x_i and for each terminal node u_j representing x_i in circuit \mathcal{C}' , select uniform random values $c_{ij\ell}$ from Z_2 and replace x_i at the node u_j with

$$r(x_i) = (c_{ij1}y_{i1} + c_{ij2}y_{i2} + \dots + c_{ij(q-1)}y_{i(q-1)}). \quad (2)$$

Let

$$G(z_1, \dots, z_h, y_{11}, \dots, y_{1(q-1)}, \dots, y_{n1}, \dots, y_{n(q-1)})$$

be the polynomial resulted from the above replacements for circuit \mathcal{C}' .

We need Lemmas 2 and 3 in the following to help estimate the success probability of the transformation.

Consider the vector space Z_2^n . For any vector $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k \in Z_2^n$, $1 \leq k \leq n$, let $\text{span}(\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k)$ denote the linear space generated by those k vectors. The following lemma follows directly from Lemma 6.3.1 of Blum and Kannan in [8].

Lemma 2. [8] *Assume that $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k$ are random vectors uniformly chosen from Z_2^n , $1 \leq k \leq n$ and $n \geq 1$. Let $\Pr[\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k]$ denote the probability that $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k$ are linearly independent. We have*

$$\Pr[\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k] > 0.28.$$

Koutis had a proof for $\Pr[\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k] > \frac{1}{4}$, which is contained in the proof for his Theorem 2.4 [18]. But some careful examination will show that there is a flaw in the analysis for $k = 3$. Nevertheless, we present a proof in the following.

PROOF. From the basis of linear algebra, we know that $\text{span}(\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k)$ has 2^k vectors and any vector in $Z_2^n - \text{span}(\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k)$ is linearly independent of $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k$. Note that $|Z_2^n| = 2^n$. Therefore,

$$\begin{aligned} \Pr[\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k] &= \Pr[\vec{v}_k \notin \text{span}(\vec{v}_1, \vec{v}_2, \dots, \vec{v}_{k-1})] \cdot \Pr[\vec{v}_1, \vec{v}_2, \dots, \vec{v}_{k-1}] \\ &= (1 - \frac{1}{2^{n-k+1}}) \cdot \Pr[\vec{v}_1, \vec{v}_2, \dots, \vec{v}_{k-1}] \\ &= \prod_{i=1}^k (1 - \frac{1}{2^{n-i+1}}) \\ &\geq \prod_{i=1}^k (1 - \frac{1}{2^i}). \end{aligned} \quad (3)$$

The last inequality holds because of $1 \leq k \leq n$. For any $1 \leq k \leq 40$, by simply carrying out the computation for the right product of expression (3), we obtain

$$\prod_{i=1}^k (1 - \frac{1}{2^i}) \geq 0.288788 > 0.28, \quad 1 \leq k \leq 40, \text{ and} \quad (4)$$

$$\left(\prod_{i=1}^{40} \left(1 - \frac{1}{2^i}\right) \right) \cdot \frac{40}{41} \geq 0.281444 > 0.28. \quad (5)$$

It is obvious that $2^i > i^2$ for $i \geq 41$. Combining this with expressions (3) and (5) yields, for any $k > 40$,

$$\begin{aligned} \prod_{i=1}^k \left(1 - \frac{1}{2^i}\right) &= \left(\prod_{i=1}^{40} \left(1 - \frac{1}{2^i}\right) \right) \cdot \left(\prod_{i=41}^k \left(1 - \frac{1}{2^i}\right) \right) \\ &\geq \left(\prod_{i=1}^{40} \left(1 - \frac{1}{2^i}\right) \right) \cdot \left(\prod_{i=41}^k \left(1 - \frac{1}{i^2}\right) \right) \\ &= \left(\prod_{i=1}^{40} \left(1 - \frac{1}{2^i}\right) \right) \cdot \left(\prod_{i=41}^k \frac{(i-1)(i+1)}{i^2} \right) \\ &= \left(\prod_{i=1}^{40} \left(1 - \frac{1}{2^i}\right) \right) \cdot \frac{40}{41} \cdot \frac{k+1}{k} \\ &\geq 0.28 \cdot \frac{k+1}{k} \\ &\geq 0.28 \end{aligned} \quad (6)$$

The complete proof is then derived from expressions (4) and (6).

Lemma 3. *For any integer matrix $\mathcal{A} = (a_{ij})_{n \times n}$, we have*

$$\text{perm}(\mathcal{A}) \pmod{2} = \det(\mathcal{A}) \pmod{2}. \quad (7)$$

PROOF. Let λ be any permutation of $\{1, 2, \dots, n\}$, and $\text{sign}(\lambda)$ be the sign of the permutation λ . Since for any integer b , $b \equiv -b \pmod{2}$, we have

$$\begin{aligned} \det(\mathcal{A}) \pmod{2} &= \left(\sum_{\lambda} (-1)^{\text{sign}(\lambda)} a_{1\lambda(1)} a_{2\lambda(2)} \cdots a_{n\lambda(n)} \right) \pmod{2} \\ &= \left(\sum_{\lambda} a_{1\lambda(1)} a_{2\lambda(2)} \cdots a_{n\lambda(n)} \right) \pmod{2} \\ &= \text{perm}(\mathcal{A}) \pmod{2}. \end{aligned}$$

It is obvious that the above lemma can be easily extended to any field of characteristic 2. We are now ready to estimate the success probability of the transformation.

Lemma 4. *Assume that the variable replacements are carried out over a field \mathcal{F} of characteristic 2 (e.g., \mathbb{Z}_2). If a given n -variate polynomial $F(x_1, x_2, \dots, x_n)$ that is represented by a tree-like circuit \mathcal{C} has a q -monomial of x -variables with degree k , then, with a probability at least 0.28^k , G has a unique multilinear monomial $\alpha\pi$ such that π is a degree k multilinear monomial of y -variables*

and α is a multilinear monomial of z -variables with degree $\leq 2k - 1$. If F has no q -monomials, then G has no multilinear monomials of y -variables, i.e., G has no monomials of the format $\beta\phi$ such that β is a multilinear monomial of z -variables and ϕ is a multilinear monomial of y -variables.

PROOF. We first show the second part of the lemma, i.e., if F has no q -monomials, then G has no multilinear monomials of y -variables. Suppose otherwise that G has a multilinear monomial $\beta\phi$. Let $\phi = \phi_1\phi_2 \cdots \phi_s$ such that ϕ_j is the product of all the y -variables in ϕ that are used to replace the variable x_{i_j} , and let $\deg(\phi_j) = d_j$, $1 \leq j \leq s$. Consider the subtree \mathcal{T}' of \mathcal{C}' that generates $\beta\phi$ when the x -variables are replaced by a linear sum of y -variables according to expression (2). Then, the subtree \mathcal{T} in \mathcal{C}^* that corresponds to \mathcal{T}' in \mathcal{C}' computes a monomial $\pi = x_{i_1}^{d_1} x_{i_2}^{d_2} \cdots x_{i_s}^{d_s}$ and ϕ is a multilinear monomial in the expansion of the replacement $r(\pi)$, which is obtained by replacing each occurrence of x -variable with a linear sum of $(q - 1)$ many y -variables by expression (2). If there is one d_j such that $d_j \geq q$, then let us look at the replacements for $x_{i_j}^{d_j}$, denoted as

$$r(x_{i_j}^{d_j}) = \prod_{t=1}^{d_j} (c_{t1}y_1 + c_{t2}y_2 + \cdots + c_{t(q-1)}y_{(q-1)}).$$

Since $d_j \geq q$, by the pigeon hole principle, the expansion of the above $r(x_{i_j}^{d_j})$ has no multilinear monomials. Thereby, we must have $1 \leq d_j \leq q - 1$, $1 \leq j \leq s$. Hence, π is a q -monomial in F , a contradiction to our assumption at the beginning. Therefore, when F has no q -monomials, then G must not have any multilinear monomials of y -variables.

We now prove the first part of the lemma. Suppose F has a q -monomial $\pi = x_{i_1}^{s_1} x_{i_2}^{s_2} \cdots x_{i_t}^{s_t}$ with $1 \leq s_j \leq q - 1$, $1 \leq j \leq t$, and $k = \deg(\pi)$. By Lemma 1, F' has at least one monomial corresponding to π . Moreover, each of such monomials has a format $\alpha\pi$ such that α is a unique multilinear monomials of z -variables with $\deg(\alpha) \leq 2k - 1$. Let $\beta = \alpha\pi$ be one of such monomials. Consider the subtree \mathcal{T}' of \mathcal{C}' that generates β . Based on the construction of \mathcal{C}' , \mathcal{T}' has s_j terminal nodes representing s_j occurrences of x_{i_j} in π , $1 \leq j \leq t$. By variable replacements in expression (2), β becomes $r(\beta)$ as follows:

$$\begin{aligned} r(\beta) &= \alpha r(\pi) \\ &= \alpha \prod_{\ell=1}^t \left[\prod_{j=1}^{s_\ell} (c_{\ell j 1}y_{\ell 1} + c_{\ell j 2}y_{\ell 2} + \cdots + c_{\ell j (q-1)}y_{\ell (q-1)}) \right], \end{aligned} \quad (8)$$

where each occurrence j of x_{i_ℓ} is replaced by $(c_{\ell j 1}y_{\ell 1} + c_{\ell j 2}y_{\ell 2} + \cdots + c_{\ell j (q-1)}y_{\ell (q-1)})$. For $1 \leq \ell \leq t$, let $\pi_\ell = x_{i_\ell}^{s_\ell}$, and

$$r(\pi_\ell) = \prod_{j=1}^{s_\ell} (c_{\ell j 1}y_{\ell 1} + c_{\ell j 2}y_{\ell 2} + \cdots + c_{\ell j (q-1)}y_{\ell (q-1)}). \quad (9)$$

Since $1 \leq s_\ell \leq q - 1$, by expression (9), $r(\pi_\ell)$ has a multilinear monomial π'_ℓ with coefficient c_ℓ such that

$$\pi'_\ell = y_{\ell 1} y_{\ell 2} \cdots y_{\ell s_\ell}, \text{ and} \quad (10)$$

$$c_\ell = \text{perm}(C[\pi'_\ell, r(\pi_\ell)]), \quad (11)$$

where the coefficient matrix, as defined in Definition 1, is

$$C[\pi'_\ell, r(\pi_\ell)] = \begin{pmatrix} c_{\ell 11} & c_{\ell 12} & \cdots & c_{\ell 1 s_\ell} \\ c_{\ell 21} & c_{\ell 22} & \cdots & c_{\ell 2 s_\ell} \\ & & \cdots & \\ c_{\ell s_\ell 1} & c_{\ell s_\ell 2} & \cdots & c_{\ell s_\ell s_\ell} \end{pmatrix}.$$

Since the field \mathcal{F} has characteristic 2 and all the entries in the coefficient are 0/1 values, we have by Lemma 3

$$\text{perm}(C[\pi'_\ell, r(\pi_\ell)]) = \det(C[\pi'_\ell, r(\pi_\ell)]).$$

Because each row of $C[\pi'_\ell, r(\pi_\ell)]$ is a uniform random vector in $Z_2^{s_\ell}$, by Lemma 2, with a probability of at least 0.28, those row vectors are linearly independent, implying $\det(C[\pi'_\ell, r(\pi_\ell)]) = 1$. Hence, by expressions (9), (10) and (11), with a probability at least 0.28, $r(\pi_\ell)$ has a multilinear monomial π'_ℓ . By expression (8), with a probability at least $0.28^t \geq 0.28^k$, $\alpha r(\pi)$ has a desired multilinear monomial $\alpha \pi'_1 \pi'_2 \cdots \pi'_t$.

5. Randomized Testing of q -monomials

Let $d = \log_2(2k-1)+1$ and $\mathcal{F} = \text{GF}(2^d)$ be a finite field of 2^d many elements. We consider the group algebra $\mathcal{F}[Z_2^k]$. Please note that the field $\mathcal{F} = \text{GF}(2^d)$ has characteristic 2. This implies that, for any given element $w \in \mathcal{F}$, adding w for any even number of times yields 0. For example, $w + w = 2w = w + w + w + w = 4w = 0$.

The algorithm RandQMT for testing whether any given n -variate polynomial $F(x_1, x_2, \dots, x_n)$ that is presented by a tree-like circuit \mathcal{C} has a q -monomial of degree k is given in the following.

Algorithm RandQMT (Randomized q-Monomials Testing):

1. As described in Subsection 4.1, reconstruct the circuit \mathcal{C} to obtain \mathcal{C}^* that computes the same polynomial $F(x_1, x_2, \dots, x_n)$ and then introduce new z -variables to \mathcal{C}^* to obtain the new circuit \mathcal{C}' that computes $F'(z_1, z_2, \dots, z_h, x_1, x_2, \dots, x_n)$.
2. Repeat the following loop for at most $(\frac{1}{0.28})^k$ times.

- 2.1.** For each variable x_i and for each terminal node u_j representing x_i in circuit \mathcal{C}' , select uniform random values $c_{ij\ell}$ from Z_2 and replace x_i at the node u_j with

$$r(x_i) = (c_{ij1}y_{i1} + c_{ij2}y_{i2} + \cdots + c_{ij(q-1)}y_{i(q-1)}). \quad (12)$$

Let

$$G(z_1, \dots, z_h, y_{11}, \dots, y_{1(q-1)}, \dots, y_{n1}, \dots, y_{n(q-1)})$$

be the polynomial resulted from the above replacements for circuit \mathcal{C}' .

- 2.2.** Select uniform random vectors $\vec{v}_{ij} \in Z_2^k - \{\vec{v}_0\}$, and replace the variable y_{ij} with $(\vec{v}_{ij} + \vec{v}_0)$, $1 \leq i \leq n$ and $1 \leq j \leq q-1$.
- 2.3.** Use \mathcal{C}' to calculate

$$\begin{aligned} G' &= G(z_1, \dots, z_h, (\vec{v}_{11} + \vec{v}_0), \dots, (\vec{v}_{1(q-1)} + \vec{v}_0), \dots, \\ &\quad (\vec{v}_{n1} + \vec{v}_0), \dots, (\vec{v}_{n(q-1)} + \vec{v}_0)) \\ &= \sum_{j=1}^{2^k} f_j(z_1, \dots, z_h) \cdot \vec{v}_j, \end{aligned} \quad (13)$$

where each f_j is a polynomial of degree $\leq 2k-1$ over the finite field $\mathcal{F} = \text{GF}(2^d)$, and \vec{v}_j with $1 \leq j \leq 2^k$ are the 2^k distinct vectors in Z_2^k .

- 2.4.** Perform polynomial identity testing with the Schwartz-Zippel algorithm [20] for every f_j over \mathcal{F} . Return "yes" if one of those polynomials is not identical to zero.

- 3.** Return "no" if no "yes" has been returned in the loop.

It should be pointed out that the actual implementation of Step 2.3 would be running the Schwartz-Zippel algorithm concurrently for all f_j , $1 \leq j \leq 2^k$, utilizing the circuit \mathcal{C}' . If one of those polynomials is not identical to zero, then the output of G' as computed by circuit \mathcal{C}' is not zero.

The group algebra technique established by Koutis [18] assures the following two properties:

Lemma 5. ([18]) *Replacing all the variables y_{ij} in G with group algebraic elements $\vec{v}_{ij} + \vec{v}_0$ will make all monomials $\alpha\pi$ in G become zero, if π is non-multilinear with respect to y -variables. Here, α is a product of z -variables.*

PROOF. Recall that \mathcal{F} has characteristic 2. For any $\vec{v} \in Z_2^k$, in the group algebra $\mathcal{F}[Z_2^k]$,

$$\begin{aligned} (\vec{v} + \vec{v}_0)^2 &= \vec{v} \cdot \vec{v} + 2 \cdot \vec{v} \cdot \vec{v}_0 + \vec{v}_0 \cdot \vec{v}_0 \\ &= \vec{v}_0 + 2 \cdot \vec{v} + \vec{v}_0 \\ &= 2 \cdot \vec{v}_0 + 2 \cdot \vec{v} = \mathbf{0}. \end{aligned} \quad (14)$$

Thus, the lemma follows directly from expression (14).

Lemma 6. ([18]) *Replacing all the variables y_{ij} in G with group algebraic elements $\vec{v}_{ij} + \vec{v}_0$ will make any monomial $\alpha\pi$ to become zero, if and only if the vectors \vec{v}_{ij} are linearly dependent in the vector space Z_2^k . Here, π is a multilinear monomial of y -variables and α is a product of z -variables. Moreover, when π becomes non-zero after the replacements, it will become the sum of all the vectors in the linear space spanned by those vectors.*

PROOF. The analysis below gives a proof for this lemma. Suppose V is a set of linearly dependent vectors in Z_2^k . Then, there exists a nonempty subset $T \subseteq V$ such that $\prod_{\vec{v} \in T} \vec{v} = \vec{v}_0$. For any $S \subseteq T$, since $\prod_{\vec{v} \in T} \vec{v} = (\prod_{\vec{v} \in S} \vec{v}) \cdot (\prod_{\vec{v} \in T-S} \vec{v})$, we have $\prod_{\vec{v} \in S} \vec{v} = \prod_{\vec{v} \in T-S} \vec{v}$. Thereby, we have

$$\prod_{\vec{v} \in T} (\vec{v} + \vec{v}_0) = \sum_{S \subseteq T} \prod_{\vec{v} \in S} \vec{v} = \mathbf{0},$$

since every $\prod_{\vec{v} \in S} \vec{v}$ is paired by the same $\prod_{\vec{v} \in T-S} \vec{v}$ in the sum above and the addition of the pair is annihilated because \mathcal{F} has characteristic 2. Therefore,

$$\begin{aligned} \prod_{\vec{v} \in V} (\vec{v} + \vec{v}_0) &= \left(\prod_{\vec{v} \in T} (\vec{v} + \vec{v}_0) \right) \cdot \left(\prod_{\vec{v} \in V-T} (\vec{v} + \vec{v}_0) \right) \\ &= \mathbf{0} \cdot \left(\prod_{\vec{v} \in V-T} (\vec{v} + \vec{v}_0) \right) = \mathbf{0}. \end{aligned}$$

Now consider that vectors in V are linearly independent. For any two distinct subsets $S, T \subseteq V$, we must have $\prod_{\vec{v} \in T} \vec{v} \neq \prod_{\vec{v} \in S} \vec{v}$, because otherwise vectors in $S \cup T - (S \cap T)$ are linearly dependent, implying that vectors in V are linearly dependent. Therefore,

$$\prod_{\vec{v} \in V} (\vec{v} + \vec{v}_0) = \sum_{T \subseteq V} \prod_{\vec{v} \in T} \vec{v}$$

is the sum of all the $2^{|V|}$ distinct vectors spanned by V .

Theorem 1. *Let $q > 2$ be any fixed integer and $F(x_1, x_2, \dots, x_n)$ be an n -variate polynomial represented by a tree-like circuit \mathcal{C} of size $s(n)$. Then the randomized algorithm RandQMT can decide whether F has a q -monomial of degree k in its sum-product expansion in time $O^*(7.15^k s^2(n))$.*

For applications, we often require that the size of a given circuit is a polynomial in n . in such cases, the upper bound in the theorem becomes $O^*(7.15^k)$.

PROOF. From the introduction of the new z -variables to the circuit \mathcal{C}' , it is easy to see that every monomial in F' has the format $\alpha\pi$, where π is a product of x -variables and α is a product of z -variables. Since only x -variables are replaced by respective linear sums of new y -variables as specified in expression (12) (or expression (2)), monomials in G have the format $\beta\phi$, where ϕ is a product of y -variables and β is a product of z -variables.

Suppose that F has no q -monomials. By Lemma 4, G has no monomials $\beta\phi$ such that ϕ is a multilinear monomial of y -variables and β is a product of z -variables. In other words, for every monomial $\beta\phi$ in G , the y -variable product ϕ must not be multilinear. Moreover, by Lemma 5, replacing y -variables will make ϕ in every monomial $\beta\phi$ in G to become zero. Hence, the replacements will make G to become zero and so the algorithm RandQMT will return "no".

Assume that F has a q -monomial of degree k . By Lemma 4, with a probability at least 0.28^k , G has a monomial $\beta\phi$ such that ϕ is a y -variable multilinear monomial of degree k and β is a z -variable multilinear monomial of degree $\leq 2k - 1$. It follows from Lemma 2, a list of uniform vectors from Z_2^k will be linearly independent with a probability at least 0.28. By Lemma 6, with a probability at least 0.28, the multilinear monomial ϕ will not be annihilated by the group algebra replacements at Steps 2.2 and 2.3. Precisely, with a probability at least 0.28, $\beta\phi$ will become

$$\lambda(\beta\phi) = \sum_{i=1}^{2^k} \beta \vec{v}_i, \quad (15)$$

where \vec{v}_i are distinct vectors in Z_2^k .

Let \mathcal{S} be the set of all those multilinear monomials $\beta\phi$ that survive the group algebra replacements for y -variables in G . Then,

$$\begin{aligned} G' &= G(z_1, \dots, z_h, (\vec{v}_{11} + \vec{v}_0), \dots, (\vec{v}_{1(q-1)} + \vec{v}_0), \dots, \\ &\quad (\vec{v}_{n1} + \vec{v}_0), \dots, (\vec{v}_{n(q-1)} + \vec{v}_0)) \\ &= \sum_{\beta\phi \in \mathcal{S}} \lambda(\beta\phi) \\ &= \sum_{\beta\phi \in \mathcal{S}} \left(\sum_{i=1}^{2^k} \beta \vec{v}_i \right) \\ &= \sum_{j=1}^{2^k} \left(\sum_{\beta\phi \in \mathcal{S}} \beta \right) \vec{v}_j \end{aligned} \quad (16)$$

Let

$$f_j(z_1, \dots, z_h) = \sum_{\beta\phi \in \mathcal{S}} \beta.$$

By Lemmas 4 and 5, the degree of β is at most $2k - 1$. Hence, the coefficient polynomial f_j with respect to \vec{v}_j in G' after the algebra replacements has degree $\leq 2k - 1$. Also, by Lemma 4, β is unique with respect to every ϕ for each monomial $\beta\phi$ in G . Thus, the possibility of a "zero-sum" of coefficients from different surviving monomials is completely avoided during the construction of f_j . Therefore, conditioned on that \mathcal{S} is not empty, F' must not be identical to zero, i.e., there exists at least one f_j that is not identical to zero. At Step 2.4,

we use the randomized algorithm by Schwartz-Zippel [20] to test whether f_j is identical to zero. It is known that this testing can be done with a probability at least $\frac{2^k-1}{|\mathcal{F}|} = \frac{1}{2}$ in time polynomially in $s(n)$ and $\log_2 |\mathcal{F}| = 1 + \log_2(2k-1)$. Since \mathcal{S} is not empty with a probability at least 0.28, the success probability of testing whether G has a degree k multilinear monomial is at least $0.28 \times \frac{1}{2} > \frac{1}{8}$, under the condition that G has at least one degree k multilinear monomial.

Summarizing the above analysis, when F has a q -monomial of degree k with a probability at least 0.28^k , G has a degree k multilinear monomial ϕ of y -variables in the format $\beta\phi$ with coefficient β that is a multilinear monomial of z -variables with degree $\leq 2k-1$. Thus, the probability that G does not have any degree k multilinear monomials of y -variables in the aforementioned format $\beta\phi$ in its sum-product expansion during any of the $(\frac{1}{0.28})^k$ loop iterations is at most

$$(1 - (0.28)^k)^{(\frac{1}{0.28})^k} \leq \frac{1}{e}.$$

This implies that the probability that G has at least one degree k multilinear monomial during at least one of the $(\frac{1}{0.28})^k$ loop iterations is at least

$$1 - \frac{1}{e}.$$

When G has at least one degree k multilinear monomial ϕ of y -variables in the format $\beta\phi$ as described above, the group algebra replacement technique and the Schwartz-Zippel polynomial identity testing algorithm as analyzed above will detect this with a probability at least $\frac{1}{8}$. Therefore, when F has one q -monomial in its sum-product expansion, with a probability at least

$$\frac{1}{8} \times \left(1 - \frac{1}{e}\right),$$

algorithm RandQMT will detect this.

Finally, we address the issues about how to calculate G' and the time needed to do so. Naturally, every element in the group algebra $\mathcal{F}[Z_2^k]$ can be represented by a vector in $Z_2^{2^k}$. Adding two elements in $\mathcal{F}[Z_2^k]$ is equivalent to adding the two corresponding vectors in $Z_2^{2^k}$, and the latter can be done in $O(2^k)$ time via component-wise sum. In addition, multiplying two elements in $\mathcal{F}[Z_2^k]$ is equivalent to multiplying the two corresponding vectors in $Z_2^{2^k}$, and the latter can be done in $O(k2^k \log_2 |\mathcal{F}|) = O(k2^k)$ with the help of a similar Fast Fourier Transform style algorithm as in Williams [22]. Calculating G' consists of $n * s^2(n)$ arithmetic operations of either adding or multiplying two elements in $\mathcal{F}[Z_2^k]$ based on the circuit C' . Hence, the total time needed is $O(n * s^2(n)k2^k) = O^*(2^k s^2(n))$. At Step 2.4, we run the Schwartz-Zippel algorithm on G' to simultaneously test whether there is one f_j such that f_j is not identical to zero. The total time for the entire algorithm is $O^*(2^k s^2(n) \cdot (\frac{1}{0.28})^k)$. Since

$$2 \times \frac{1}{0.28} = 2 \times \frac{100}{28} < 7.15,$$

the time complexity of algorithm RandQMT is bounded by $O^*(7.15^k s^2(n))$.

6. Concluding Remarks

The group algebra approaches to testing multilinear monomials [18, 22] and q -monomials for prime q [13, 12] rely on the property that Z_2 and Z_q are fields for primes $q > 2$. These approaches are not applicable to the general case of testing q -monomials, since Z_q is no longer a field when q is not prime. In this paper, we have developed a variable replacement technique and a new way to reconstruct a given circuit. When the two are combined, they help us transform the q -monomial testing problem to the multilinear monomial testing problem in a randomized setting. We have also proved that the transformation has the desired success probability to warrant its application to the design of our new algorithm.

It should be pointed out that the time complexity of the randomized q -monomial testing algorithm obtained in [10] runs in time $O^*(q^k)$ for prime $q \geq 2$, when the size of the circuit is a polynomial in n . Algorithm RandQMT runs in time $O^*(7.15^k)$, hence it significantly improves the time complexity of the algorithm in [10] for prime $q > 7$.

Acknowledgments

Shenshi is supported by Dr. Bin Fu's NSF CAREER Award, 2009 April 1 to 2014 March 31. Yaqing is supported by a UTPA Graduate Assistantship. Part of Quanhai's work was done while he was visiting the Department of Computer Science at the University of Texas-Pan American.

References

- [1] Hasan Abasi and Nader Bshouty, A simple algorithm for undirected hamiltonicity, ECCC, Report No. 12, 2013.
- [2] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena, PRIMES is in P, Ann. of Math, 160(2): 781-793, 2004.
- [3] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan and Mario Szegedy, Proof verification and the hardness of approximation problems, Journal of the ACM 45 (3): 501-555, 1998.
- [4] Bengt Aspvall, Michael F. Plass and Robert E. Tarjan, A linear-time algorithm for testing the truth of certain quantified boolean formulas, Information Processing Letters 8 (3): 121-123, 1979.
- [5] Richard Beigel, The polynomial method in circuit complexity, Proceedings of the Eighth Conference on Structure in Complexity Theory, pp. 82-95, 1993.
- [6] A. Björklund, Determinant sums for undirected hamiltonicity, Proceedings of the 51th IEEE Foundations of Computer Science (FOCS'2010), pp. 173-182, 2010.

- [7] A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, Narrow sieves for parameterized paths and packings. *arXiv:1007.1161v1*, 2010.
- [8] M. Blum, S. Kannan, Designing programs that check their work. *J. ACM* 42(1):269-291, 1995.
- [9] Zhixiang Chen and Bin Fu, Approximating Multilinear Monomial Coefficients and Maximum Multilinear Monomials in Multivariate Polynomials. *J. Comb. Optim.* Forthcoming. On-line completion: May-2012. DOI Information: 10.1007/s10878-012-94965. 2012.
- [10] Zhixiang Chen and Bin Fu, The Complexity of Testing Monomials in Multivariate Polynomials, Proceedings of the Fourth International Conference on Combinatorial Optimization and Applications (COCOA'2011), LNCS 6831, pp. 1-15, August 2011.
- [11] Zhixiang Chen and Bin Fu, Approximating multilinear monomial coefficients and maximum multilinear monomials in multilinear polynomials, Proceedings of the Fourth International Conference on Combinatorial Optimization and Applications (COCOA'2010), LNCS 6508, pp. 309-323, 2010.
- [12] Zhixiang Chen, Bin Fu, Yang Liu, and Robert T. Schweller, On Testing Monomials in Multivariate Polynomials. *Journal of Theoretical Computer Science*, Forthcoming. On-line completion: 13-APR-2012. DOI information: 10.1016/j.tcs.2012.03.038. 2012.
- [13] Zhixiang Chen, Bin Fu, Yang Liu, and Robert T. Schweller, Algorithms for Testing Monomials in Multivariate Polynomials, Proceedings of the Fifth International Conference on Combinatorial Optimization and Applications (COCOA'2011), LNCS 6831, pp. 16-30, 2011.
- [14] Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra and Mario Szegedy, Interactive proofs and the hardness of approximating cliques, *Journal of the ACM (ACM)* 43 (2): 268292, 1996.
- [15] Bin Fu, Separating PH from PP by relativization, *Acta Math. Sinica* 8(3):329-336, 1992.
- [16] Valentine Kabanets and Russell Impagliazzo, Derandomizing polynomial identity tests means proving circuit lower bounds, *STOC*, pp. 355-364, 2003.
- [17] Adam Klivans and Rocco A. Servedio, Learning DNF in time $2^{\tilde{O}(n^{1/3})}$, *STOC*, pp. 258-265, 2001.
- [18] Ioannis Koutis, Faster algebraic algorithms for path and packing problems, Proceedings of the International Colloquium on Automata, Language and Programming (ICALP), LNCS, vol. 5125, Springer, pp. 575-586, 2008.

- [19] Marvin L. Minsky and Seymour A. Papert, *Perceptrons* (expanded edition 1988), MIT Press, 1968.
- [20] Rajeev Motwani and Prabhakar Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995.
- [21] Adi Shamir, $IP = PSPACE$, *Journal of the ACM*, 39(4): 869-877, 1992.
- [22] Ryan Williams, Finding paths of length k in $O^*(2^k)$ time, *Information Processing Letters*, 109, 315-318, 2009.